WIDEBAND GAPFILLER SATELLITE SYSTEM

RFP ANNEX M

SOFTWARE DEVELOPMENT CAPABILITIES EVALUATION (SDCE) QUESTIONNAIRE

SOFTWARE DEVELOPMENT CAPABILITY EVALUATION (SDCE)

In order to assure that offerors have the software development capabilities required for the Wideband Gapfiller program, the Government will conduct a Software Development Capability Evaluation. The SDCE will be conducted with the prime offeror and proposed team members and subcontractors who have significant software development responsibility. This evaluation will be based on an analysis of the following documentation that is to be submitted with the offerors' proposals. For instances of teaming and prime/subcontractor arrangements among offerors, it is the responsibility of the prime offeror to determine the required information (such as proposal information, SDCE question responses, and supporting data) that is to be supplied to the Government by each member of the bidding team.

The offeror shall submit an electronic media copy of the SDCE as part of the proposal submission on CD-ROM. The offeror shall also provide three paper copies of the SDCE. If electronic versions of supporting data are not available, the offeror may submit paper copy only for that piece of data.

The following information in direct support of the SDCE is to be submitted with the proposal and will not be limited by the specified page counts for the proposal:

1.0 OVERVIEW OF SOFTWARE DEVELOPMENT EFFORT

The offeror shall provide an overview for the total software development effort that addresses: the organization of the offeror's team; the task and responsibility distribution among the team members; and the processes used to manage and control team member performance. The overview should address the total Wideband Gapfiller life cycle. The purpose of this overview is to provide a foundation for review of the SDCE questionnaire responses. All information in this overview shall be consistent with information provided in the planning documentation and other volumes of the RFP response, and shall reference such information where appropriate. Specific volume, page, and paragraph numbers are required where references are used in the SDCE responses. The overview shall be limited to 8 pages.

2.0 RESPONSES TO QUESTIONS

Responses to the questions (see tailored questions in attachment 2) are encouraged to be provided directly in the documentation accompanying the proposal, such as the IMP or IMS, or other proposal volumes. When responses to the SDCE questions are provided in the documentation accompanying the proposal, specific page number and paragraph references shall be provided with the response to the question. This approach is intended to reduce the SDCE preparation effort and eliminate duplication within the proposal. Responses should be concise and unambiguous, not exceeding two pages per question. Responses should be provided for the processes to be employed on the Wideband Gapfiller program by the offeror and any team members or subcontractors who will provide a major portion of the software. Common processes among the team members require only one response. However, for processes not common among team members, the combined responses shall not exceed the page limits specified above. The response to one question may refer to the response to another, when appropriate. Each question shall reference supporting data that defines the process or provides evidence of implementation; this may be done in the question response or in a separate cross-reference matrix.

3.0 SUBSTANTIATING DOCUMENTS FOR EXISTING PROCESSES.

Substantiating documents must be submitted for all existing processes that are planned for use on the Wideband Gapfiller development, whether employed by the prime offeror, team members, or subcontractors. Results of previous SDCEs or SCEs will not be considered. Examples of substantiating documents include:

- Copies of corporate software-related procedure, process, standard, and practice descriptions that are relevant to the acquisition. (Also for each subcontractor and team member if different procedures, processes, or practices are to be employed.)
- Copies of documents that provide evidence of use of the proposed processes (e.g., development schedules, software development plans, software requirements specifications, test and integration plans, and procedures). Documents provided as evidence of use of the proposed processes may be obtained from another project or from the current project. Supporting data should be submitted for only one additional representative software development effort besides the current effort. Sample data from more than one additional program is not necessary unless the representative program does not have the specific document or data type.

4.0 DESCRIPTION OF NEW PROCESSES.

For new processes not yet documented, describe the benefits and risks of using the new process and the rationale for employing them in lieu of examples of past application. This description will not count against the SDCE response page limits.

5.0 DATA INDEX AND SAMPLE DATA COVER SHEET.

The following forms must be completed and submitted with the proposal: (This data will not count against proposal page limits.)

- An index of all supporting material submitted along with a reference to where that supporting material can be found.
- Cover Sheet for Project Sample Data for each sample submitted (see attachment #1).
- Cross references between the questions in the CC and the specific portions of the sample material submitted that answer the questions or provide evidence of their implementation. Cross references between the questions in the SDCE and the specific portions of the sample material submitted that answer the questions or provide evidence of their implementation. If specific references are not provided (by page number and/or paragraph number(s)), the referenced evidence will not be considered in the evaluation. References to evidence that must be accessed via the Internet or an intranet will not be considered in the evaluation.

The format and method of providing the index and cross-references is at the discretion of the offeror.

Attachments:

1. Cover Sheet for Project Sample Data (Attachment 1)
2. Tailored SDCE Questions (Attachment 2)

Attachment 1: Sample Data Cover Sheet

This attachment contains an example that illustrates how to complete the Sample Data Cover Sheet and one copy of a blank Sample Data Sheet.

EXAMPLE

Cover Sheet for Project's Sample Data
Contractor: Team A
Sample Project Name: Project X
Sample Project Customer: U.S. Air Force Space and Missile Systems Center
Critical Capability(ies): 4.4.2 Metrics Application
Title of Sample: Project X Software Development Metrics Reports
Explain why your experience on the sample project is relevant to the proposed project. Object-oriented methods and metrics were used on the sample project. The same object-oriented methods and metrics are planned for use on the proposed project.

ATTRIBUTES	**PROPOSED PROJECT**	**SAMPLE PROJECT**
Application Domain	Weather Satellite	Communications Satellite
Product Type	Ground System (Command and Control)	Ground System (Command and Control)
Acquisition Phase[1]	EMD	EMD
Software Development Phase(s)	Design; Coding and Unit Test	Coding and Unit Test, Increments 1 and 2
Award Date		1/94
Contract Duration	8 Years	5 Years
Current Project Phase/ Contract Month[2]		EMD: Between System PDR and System CDR/Month 24
Prime/Subcontractors[3]	2 Software Subs	Prime & 1 Software Sub
Software KSLOC[4]	750	500
Language(s) and Percentages	Ada 95: 90% C++: 10%	FORTRAN 77: 75% C++: 25 %
Target Processor(s)/OS(s)	RISC 6000/UNIX	VAX 6200/VMS 6.2
Applicable Standards	IEEE 1498	DoD-STD-2167A & 2168

[1] For "Proposed Project," phase(s) in which Critical Capability(ies) are to be used; for "Sample Project," phase in which sample was generated.

[2] Phase/month of the Sample Project as of the current date.

[3] Contractors developing the software products specified in the "Product Type" row

[4] Total number of KSLOC for software specified in the "Product Type" row

Cover Sheet for Project's Sample Data

Contractor:

Sample Project Name:

Sample Project Customer:

Critical Capability(ies):

Title of Sample:

Explain why your experience on the sample project is relevant to the proposed project.

ATTRIBUTES	PROPOSED PROJECT	SAMPLE PROJECT
Application Domain		
Product Type		
Acquisition Phase[1]		
Software Development Phase(s)		
Award Date	▨	
Contract Duration		
Current Project Phase/ Contract Month[2]	▨	
Prime/Subcontractors[3]		
Software KSLOC[4]		
Language(s) and Percentages		
Target Processor(s)/OS(s)		
Applicable Standards		

[1] For "Proposed Project," phase(s) in which Critical Capability(ies) are to be used; for "Sample Project," phase in which sample was generated.

[2] Phase/month of the Sample Project as of the current date.

[3] Contractors developing the software products specified in the "Product Type" row

[4] Total number of KSLOC for software specified in the "Product Type" row

Attachment 2: Questionnaire
The hierarchy of Functional Areas, Critical Capability Areas, and the Critical Capabilities that have questions for the Wideband Gapfiller program is below. The specific questions for the Wideband Gapfiller SDCE follow.

1. Program Management
1.2. Program Planning and Tracking
1.2.1 Planning
1.2.4 Schedules
1.3. Subcontractor Management
1.3.2 Subcontractor Development Management
1.5. Risk Control
1.5.1 Risk Identification

2. Systems Engineering
2.1. System Requirements Development, Management, and Control
2.1.5 Requirements Traceability
2.4. Intergroup Coordination
2.4.1 Group Interfaces
2.5. Systems Engineering Planning
2.5.1 Methodology and Standards
2.6. System Integration and Test
2.6.1 Integration and Test Planning
2.7. Reuse
2.7.4 COTS/Reuse Software Evaluation, Selection and Management

3. Software Engineering
3.3. Software Requirements Management
3.3.1 Software Requirements Analysis
3.3.2 Software Requirement Changes
3.4. Software Design
3.4.1 Design Methodology
3.5. Software Coding and Unit Testing
3.5.1 Code Development
3.5.2 Code Changes
3.6. Software Integration and Test
3.6.1 Software Integration
3.6.2 Software Testing

4. Quality Management and Product Control
4.2. Software Quality Assurance (SQA)
4.2.3 Compliance Checking
4.5. Peer Reviews
4.5.1 Peer Review Planning
4.7. Software Configuration Management (SCM)
4.7.1 Planning
4.7.2 Baseline/Configuration Identification and Management

5. Organizational Resources and Program Support
5.6. Organizational Process Management
5.6.1 Process Planning and Coordination
5.7. System/Software Engineering Environment (S/SEE)
5.7.2 S/SEE Components

6. Program Specific Technologies
6.4 Database Management

6.4.2 Database Design
6.4.5 Database Quality Assurance

SDCE Questionnaire

Question numbers are consistent with AFMCPAM 63-103. Revised questions/criteria are identified with a (a), i.e., Q3a instead of Q3 and C3a instead of C3. The criteria that the questions are normally evaluated against are included at the end of each Critical Capability.

1.2.1

Q1 How is your software development planning integrated with systems management and hardware management? **C1a C3a**

Q3a Describe your technical and management reviews used to control the development progress throughout the entire development period. Define these events and corresponding criteria. How are these events incorporated into program documents and into which program documents are they incorporated (e.g., the IMP and IMS)? **C3a C4a**

Q5 Identify the software tracking metrics to be used on this program. Describe your process for monitoring critical status metrics or indicators. How do you determine when management action is required? Describe the conditions that would result in management action for each established metric or indicator. **C6a**

C1a The program planning adequately accounts for the integration of software development and management with system and hardware management. **Q1**

C3a The program planning is adequately documented and includes the necessary reviews, accountability, status assessment, schedule control and reporting to manage the software related system development activities leading to the definition of the software requirements baseline. **Q1 Q3a**

C4a The program planning includes an adequate series of technical and management reviews with associated completion criteria (including quality gates) that are used to control the development progress. **Q3a**

C6a Adequate metrics and metrics processes are in place and variance thresholds are established for critical status metrics (e.g., size, cost, effort, progress, and schedule). **Q5**

1.2.4

Q1 Describe your approach to establishing the software development schedules from the top system level schedule to the lowest level detail schedules. Explain how incremental (block, build) software development schedules are established. **C1**

Q5a Describe your method for monitoring and statusing software development schedules. Who is responsible for this function? Which level of schedule that addresses software is used as the baseline to track and report status? **C1**

C1 Software schedules are established in sufficient detail to maintain visibility and control of the development process including the establishment of any planned blocks, builds or increments. **Q1 Q5a**

1.3.2

Q1a Fully describe your process for subcontractor management including reporting and control of the subcontractor software development activities. How does this process relate to and integrate with your overall system program management approach? Describe how the subcontractor management and review activities are reflected in the program level documents. **C1 C3a**

C1 The proposed subcontractor management process is integral to the system program management process and provides integrated reporting and control of the subcontractor software development activities consistent with the program's management control system. **Q1a**

C3a Periodic management and technical reviews to address subcontractor development progress are conducted and are reflected in program documents such as the SDP. **Q1a**

1.5.1

Q1a Describe your process to identify and reduce risks associated with the system and software development throughout the program lifecycle. **C1a**

Q2 Identify the projected risks and shortfalls associated with this program as a result of applying this process. **C1a**

C1a An effective process is defined and is used to identify the shortfalls and risks associated with the proposed development activities, and effective means are being employed to mitigate the significant identified risks. **Q1a Q2**

2.1.5

Q1a Describe the process used to provide two-way requirements-to-requirements, requirements-to-design and requirements-to-verification traceability throughout the system life cycle. At what point are requirements-to-requirements, requirements-to-design and requirements-to-verification traceability established and documented? What provisions exist to maintain the traceability? **C1a**

C1a Two-way requirements-to-requirements, requirements-to-design and requirements-to-verification traceability are effectively maintained from system specifications to hardware and software configuration item specifications, from specifications to design documentation, and from specifications to verification planning and execution, and the information is effectively shared and used. **Q1a**

2.4.1

Q1 Describe the processes to be followed to have users' and maintainers' needs and viewpoints adequately reflected in system requirements throughout development. **C1**

Q2 Describe the processes to be followed to keep system requirements in balance with acquisition organization resources throughout development. **C1**

Q7 How will interfaces between the various system and subsystem developers be managed? **C2**

C1 Throughout the development lifecycle there is periodic coordination among developers, acquisition organizations, users, maintainers, and testers regarding user needs, acquisition organization resources, technology status, and system requirements. Requirements changes resulting from interaction with users, maintainers, and testers are managed with acquisition organization approval. **Q1 Q2**

C2 There is a systems engineering process which (as appropriate) emphasizes an integrated product development approach and defines systems engineering interfaces with other engineering disciplines and development activities, as well as interfaces between system and subsystem developers. **Q7**

2.5.1

Q1a Describe how the program's systems engineering policies, practices, procedures and standards are defined and documented and how they relate to the corresponding software systems engineering policies, practices, procedures and standards. **C1a**

C1a Effective systems engineering policies, practices, procedures and standards are defined, are consistent with systems engineering contractual standards. Effective policy, practices, procedures and standards integration exists among the systems engineering and software systems engineering organizations. **Q1a**

2.6.1

Q4a Describe any integration and test plans developed specifically for COTS software or other reuse software. **C4a**

C4a Any use of commercial-off-the-shelf (COTS) software or other reuse software is adequately incorporated into system integration and test planning. **Q4a**

2.7.4

Q1a Describe your process for evaluating and selecting COTS and reuse software, including the criteria that each product must meet before it is considered for inclusion in a development effort. **C1a, C2a**

Q2a What is your plan for managing COTS and reuse software? **C2a, C3a, C4a**

Q3a Describe how your software configuration management plan includes the configuration control of COTS and reuse software products selected for use on this program. **C5a**

C1a The contractor has a well-defined process for COTS and reuse software selection that includes effective criteria to ensure that the selected products provide needed capabilities and meet system and software constraints. **Q1a**

C2a The contractor has considered the system life cycle costs in the evaluation, selection and management of COTS and reuse software. **Q1a, Q2a**

C3a The contractor has an effective plan for managing COTS and reuse software. The plan is appropriately integrated with the software engineering management plan and systems engineering management plan. **Q2a**

C4a The COTS and reuse software management plan adequately covers technical issues, such as supportability, security, and safety; and fault isolation and detection. **Q2a**

C5a The contractor's software configuration management plan adequately incorporates processes for installing COTS and reuse software on numerous systems, managing the configuration of multiple baselines, and controlling the licensing of COTS and reuse software products. **Q3a**

3.3.1

Q1a Describe the software requirements analysis process (es) to be applied. Identify the specific methodologies and tools to be used to support the analysis process. **C3a**

C3a The selected requirements analysis methodology/methodologies is/are appropriate for the development effort, and compatible with other methodologies applied on the program. The analysis methodology is supported with necessary tools. **Q1a**

3.3.2

Q1 Describe the software development activities that result from a change in or addition to the requirements. When do they get performed? How do you ensure that they are performed? **C1a**

C1a The software development artifacts (e.g., requirements, design, code, and documentation) are appropriately revised as changes to the requirements are incorporated. **Q1**

3.4.1

Q1a Describe the process(es) and specific methodologies used to develop the software design. Describe how the methodologies interact with the requirements process, are used to maintain the design through development and are used for life cycle support. What tools are used to support the methodologies? **C1a**

Q5a How does the design methodology describe the interfaces between the software and other components of the system? **C3a**

C1a Effective methodologies are used to develop, document and maintain the software design and interface with requirements processes. The methodologies are effectively supported by tools. **Q1a**

C3a The design description includes all software interfaces with other system components. **Q5a**

3.5.1

Q10 What exit criteria exist for establishing that each lowest-level software unit is ready for integration? Do they include compliance with coding standards? Do they include peer reviews? Do they include unit testing? Do they include conformance to the design? How are they enforced? **C5a**

C5a Effective exit criteria exist for establishing that each lowest-level software unit has been implemented correctly, is performance tested, and conforms with the coding standards. **Q10**

3.5.2

Q2a Describe your process for estimating the effect of code changes on other parts of the system. What tools are used? Who is involved in the process? **C2a**

C2a Code changes are effectively reviewed for correctness, and to avoid undesired impact on other software and system components. **Q2a**

3.6.1

Q1a Describe your process for planning the software integration. How do you determine the order for integrating the different software components? Describe how your integration process accommodates all levels of software integration, how integration changes are handled and how software integration processes support hardware/software integration. **C1, C2, C4a**

C1 The software integration planning takes into account the interdependencies between the different software components and the criticality of each component. **Q1a**

C2 The software integration planning takes into account the availability of other components of the system. **Q1a**

C4a The software integration planning and process effectively accommodate software integration at all levels, effectively incorporate integration changes and support hardware/software integration. **Q1a**

3.6.2

Q1a How are verification plans, verification procedures and verification cases developed? When? By whom? Where are they documented? How are they reviewed? How are they controlled? **C1a**

Q2a What tools will be used for verification? When will they be available? Will they require any special inputs? Will their outputs require any special processing? What is your process to ensure that all required verification resources have been planned and allocated as well as qualified for use? **C2a**

Q3a Does your software test and verification process define specific levels of software test? What are they? **C3a**

Q5 Describe your process for incorporating changes resulting from errors uncovered during testing. Where is it documented? How is it enforced? **C4a**

C1a The software test process includes development of test plans, procedures, and test cases by an identified organization. These documents are produced in a timely manner and are adequately reviewed and controlled. **Q1a**

C2a A process exists to ensure that software verification is adequately planned with sufficient verification resources and that those verification resources are adequately qualified for their intended use. **Q2a**

C3a An effective approach is used that plans for all levels of testing necessary to ensure thorough testing of the software. **Q3a**

C4a An effective process exists for incorporating changes resulting from software testing. **Q5**

4.2.3

Q2a Describe how SQA ensures compliance of the software development activities with the defined processes and how discrepancies are resolved. Which processes are audited? How often? **C2a**

C2a Adherence to the defined software development and management processes is verified and discrepancies are monitored until corrected. **Q2a**

4.5.1

Q1a Describe the documented internal peer review procedures and requirements, including products that require peer reviews, definition of required participants, completion criteria and review content and follow-on action item resolution. **C1a**

C1a Internal documents exist that: identify appropriate required participants in the reviews, provide adequate specific criteria for successful completion, describe adequate documentation required for the review and describe how follow-on actions are adequately documented, tracked and controlled. **Q1a**

4.7.1

Q2a Explain how SCM is integrated with system configuration management, engineering management, and other development disciplines. **C2**

C2 The SCM is integrated with other configuration management plans used on the program **Q2a**

4.7.2

Q1a How are software and database baselines, both formal and informal, controlled using documented procedures for software and documentation and for transfer to other libraries or programs, where appropriate? **C1a**

Q5a What is the program approach to establishing and controlling formal and informal developmental baselines and verification configurations? **C4a**

C1a The configuration control implementation establishes a developmental configuration for each software and database item; effectively controls the preparation and dissemination of changes to the master copies of deliverable software, databases and documentation; and maintains current copies of deliverable documentation, databases and code. **Q1a**

C4a Effective procedures exist and are followed to create and maintain formal and informal developmental and verification baselines. **Q5a**

5.6.1

Q3a Describe the coordination of the system development and software development process management activities of the organization and the responsible individuals or groups. How are these activities coordinated with the program? How is software process compliance enforced? **C2a**

C2a The system and software process management activities of the organization are effectively coordinated and enforced. In particular, these activities include:

- Defining and managing changes to the organization's system and software processes,
- Collecting and maintaining data on use of the organization's system and software processes,
- Directing feedback to management on the program's software process activities to ensure compliance and effective use. **Q3a**

5.7.2

Q3a Describe how the tools in the S/SEE support the software development process functions and methodologies selected for the program and the relationship of the S/SEE to the life cycle maintenance environment. **C1a**
C1a The S/SEE components effectively support the program's software engineering development and management requirements, functions, methodologies, and activities, and will effectively support the maintenance environment when operational. **Q3a**

6.4.2

Q6a How will system integrators and end users be incorporated into the database design process? **C4**
C4 The database design is developed interactively with system integrators and end users. **Q6a**

6.4.5

Q7 What configuration control procedures are established for the database? **C2a**
C2a Adequate database configuration control procedures have been identified and documented. **Q7**